Pulse-Length Determination Techniques in the Rectangular Single Event Transient Fault Model

Alireza Rohani*, Hans G. Kerkhoff*, Enrico Costenaro†, Dan Alexandrescu†,
*Testable Design and Test of Integrated Systems Group
CTIT, University of Twente, Enschede, The Netherlands
Emails:a.rohani@utwente.nl, h.g.kerkhoff@utwente.nl
†iRoC Technologies, Grenoble, France
Emails:enrico@iroctech.com,dan@iroctech.com

***Abstract*—One of the well-known models to represent Single Event Transient phenomenon at the logic-level is the rectangular pulse model. However, the pulse-length in this model has a vital contribution to the accuracy and validity of the rectangular pulse model. The work presented in this paper develops two approaches for determination of the pulse-length of the rectangular pulse model used in Single Event Transient (SET) faults. The first determination approach has been extracted from radiation testing along with transistor-level SET analysis tools. The second determination approach has been elicited from asymptotic analytical behaviour of SETs in 45–nm CMOS process. The results show that applying these two pulse-length determination approaches to the rectangular pulse model will cause the fault injection results converge much faster (up to sixteen times), compared to other conventional approaches.**

I. Introduction

A very first concern with regard to Single Event Effects (SEEs) emerged during the 1990s when several experiments repeatedly showed that more than 30% of system failures are due to SEEs, rather than conventional permanent faults [1]. SEEs appear as a data corruption in the sequential or the combinational parts of a circuit. The contribution of SEEs in the combinatorial logic might lead to a momentary voltage pulse at the output of the logic, a so called Single Event Transient (SET) [3].

Recently, there have been many industrial domains in which the reliability of computer systems is the crucial factor for the whole system's reliability, such as in the automotive industry (full-hybrid and drive-free cars) and medical instruments (electronic pills). As a consequence, there is a great interest to anticipate the sensitivity of particular ICs in the presence of SETs. Fault injection has long been recognized to be a particularly attractive method to assess the vulnerability of a system with regard to SETs [4]. Fault-injection assesses the vulnerability of a system by speeding up the occurrence rate of SETs. It can be carried out at different abstraction levels, including *radiation-testing*, *simulation-based* and *emulation-based* fault injections [5].

Radiation-testing is carried out by stressing the actual hardware with real environmental parameters. This method is similar to the real physical nature of SEEs, but conducting such experiments is very complex and costly. In the *simulation-based* fault injections, SEEs are simulated in the net-list of a circuit. This method is a very useful experimental way to assess the degree of vulnerability of a system, while the system is still under development. Moreover, this method provides a very high degree of controllability over where and when faults are injected. Furthermore a high degree of observability over the propagation of an injected fault is achieved. Finally, *emulation-based* fault injections which have recently been introduced [6], can combine the flexibility and controllability of *simulation-based* techniques with the speed of *radiation-based* ones. However, the system under analysis must be fully synthesisable which confines the usage of benchmarks in the experiments.

Considering the above-mentioned categories, *simulation-based* fault injections have raised the attention in the academic community as well as in the industrial world [4]. One of the well-known models used in such fault injections in order to imitate the affect of SETs is the rectangular pulse model (also known as double exponential model) [7]. However, there is still no consistency over choosing a pulse-length which accurately imitates the real behaviour of a SET in the simulated circuit. The key contribution of this paper is developing two approaches for determining the pulse-length for the rectangular SET fault model. The first determination approach has been extracted based on the results of radiation-testing for a 45–nm gate-level library along with a precise transistor-level SET analysis tool. This determination technique has several advantages over conventional SET models; Because it takes into account the contribution of each cell in the net-list to determine the pulse-length, moreover a weighted probability of different pulse-lengths is assigned to each cell. The second pulse-length determination approach uses the asymptotic analytical behaviour of the SPICE representation of SETs to determine a pulse-length. This model takes into account the run-time activities of the strike node in the fault injection process.

In order to assess the accuracy of the enhanced representations, they have been applied to a gate-level net-list of a DSP processor (post-synthesised, including timing information). The first SET fault model constitutes of the rectangular pulse model along with the first pulse-length representation. The second SET model consists of rectangular pulse model in which its pulse-length is determined by the asymptotic analytical behaviour of the SPICE representation of SETs in a 45–nm CMOS process.

The rest of this paper has been organized as follows: section II states some previous works dealing with SET fault models. Section III introduces the first pulse-length representation derived from laser-experiments along with a transistor SET-

This research has been conducted within the ENIAC project ELESIS (296112) which is financially supported by Agentschap NL.

978-1-4799-0103-6/13/$31.00 ©2013 IEEE

analysis tool. Section IV describes the details of developing the analytical-based pulse-length determination technique. In section V, the architecture of our case study will be briefly explained. The results of the fault injection experiments for rectangular pulse model with different pulse-length representations is shown in Section VI. The assessment of different fault models in terms of accuracy and convergence is presented in section VII. Finally section VIII concludes the paper.

II. Conventional Determination of Pulse-Length in the Rectangular SET Model

A considerable amount of literature has been published to explore the logic SET model. As a definition, a SET is a momentary corruption of the voltage that would appear at a random time and lasts for a brief period of time. The above mentioned characteristics are known as *time-instant* and *pulse-length*, respectively [6].

There is a well-accepted model for circuit level representation of a SET pulse, i. e. the double exponential pulse model ([7], [8]). However, this pulse might have different lengths. To date, it is very hard to find a unanimous pulse-length in literature. For example, the authors in [9] define the pulse-length as the time between the *time-instant* and the end of an experiment. The authors in [1] and [10] interpret a SET as a momentary pulse with a random *time-instant* and a fixed pulse-length. The first work deals with an older technology in which the length of a pulse is in the order of hundreds of nanoseconds, while the latter work determines the pulse-length in a newer CMOS processes in the order of hundreds of picoseconds. None of the above mentioned models take into account the contribution of different cells in the net-list. A recent work published in [11] determines a fixed pulse-length for each cell in the net-list. We will show that a more realistic determination technique is to consider a weighted probability of different pulse-lengths for each cell. To name two works that consider system clock-period in pulse-length determination, one can mention [12] in which the pulse-length is determined from the *time-instant* until the start of the next succeeding clock edge. As a recent work, one can mention [6] where the authors define the pulse-width equal to the system clock-period.

Some studies have introduced weighted values to represent the pulse-length in collaboration with the system clock-period. For example, in [13] and [14], the pulse-lengths are selected in such a way that all those numbers form an exponential distribution function with a mean value (μ) as close as possible to the system clock-period. So, different pulse-lengths are being considered for a SET. This model takes into account the system clock-period, but not the contribution of each cell. Recently, dedicated tools such as the TFIT [15] and HSECT-SPI [10] have been specifically developed to predict and improve the cell Soft Error Rate (SER) performance at the circuit-level. These tools use a SPICE model (transistor-level) of a cell which is considered as the most accurate level to represent a SET. While it perfectly fits for SET analysis purposes, a transistor-level description is cumbersome, difficult to generate for large circuits and cannot be fully simulated with reasonable efforts. A more practical approach consists in using a Gate-Level Net-list (GLN, in a Verilog/VHDL format) complemented by timing information (e.g. SDF - Standard Delay Format files). However, SETs have to be represented in a logic model in order to be used with these descriptions.

Our first determination technique of SETs in the logic model (here-after named: *circuit-based model*) uses the well-known rectangular pulse model in which its pulse-length has been extracted based on the results of the SER characterization for a complete standard cell library (via TFIT). The second model (to be referred as: *analytical-based model*) uses the most-common SPICE-level representation of a SET in order to develop a simplified pulse-length.

III. Circuit-Based Pulse-Length Determination Technique

In order to extract the pulse-length from the real physical experiments, the first step consists in characterizing the electrical effects induced by energetic particles in a standard cell library. This characterization is performed using a transistor implementation of each cell in in the library. The library is the 45–nm open-access NANGATE repository [16]. The circuit-implementation of the library and real measurements have been gathered in a SER database to build a logic fault model dedicated to each cell. As our ultimate aim is to determine a SET logic model for each cell, the following parameters with regard to the SET have been investigated:

- The Pulse-Length (PL) of the SET
- The Soft Error Rate (SER) of a SET with the specific PL

The above-mentioned information will be unique for each cell in the standard cell library. A complete characterization will produce a $Cell_{(PL,SER)}$ for all cells in the library.

In order to find the (PL, SER) associated with each cell, an EDA tool can be used. Dedicated tools such as TFIT [15] from iRoC Technologies represents a new generation of tools that allows reasonably accurate calculation of the impact of electrical effects of particles with respect to a transistor or a cell. A vast number of experiments have been carried out to specify a complete database for all possible $Cell_{(PL,SER)}$ in the 45–nm NANGATE library. The detailed explanation of the conducted experiments is out of the scope of this paper, as our aim is to use the outcome of those experiments to develop a pulse-length to build a realistic logic SET-pulse model. A detailed explanation of those experiments has been published in a separate paper [17].

The TFIT tool works by pre-characterizing the cell with regard to SETs. Different SETs are injected and simulated in all the relevant nodes of a cell. The expected outcome is a SET with the given minimal duration. Then, the tool analyses the working environment. TFIT uses a Nuclear Database to evaluate any possible secondary particle produced by an atomic reaction between a neutron and the silicon atoms. Direction and energy of those secondary particles are studied to account their interaction with the sensitive volumes of the cell (previously computed by the tool). Depending on the type of interaction, a current is injected while the output of the cell is monitored to observe any possible electrical event, during the analysis, the Electrical Masking is implicitly considered: transient pulses too weak ($V \leq V_{dd}/2$) are discarded. The tool also uses a technological SER process response model (in this case a 45nm generic), that is a database where a collection of relevant (to a given process technology) current pulses are stored. Theses currents are the ones used to perform

TABLE I: SER expressed in FIT for three cells in the NANGATE Library

Cell	50ps	75ps	100ps	125ps	150ps	175ps
AND2	51.1	29.88	9.39	0.30	0	0
NAND2	41.6	20.6	2.8	0.02	0	0
XNOR2	69.6	60.2	30.41	2.0	0.05	0

the analysis.

An intermediary step of the analysis consists of the calculation of the cross-section corresponding to a sensitive volume within the device corresponding to the specific particle. By cross-checking the possible environment-induced events versus the data recorded during the pre-characterization, the TFIT tool is able to compute the Soft Error Rate (SER) value, expressed in FIT (Failure In Time) for each pulse-length value.

The specific pulse-length values have been defined as 50ps, 75ps, 100ps, 125ps, 150ps and 175ps. The probability of occurrence of a pulse-length smaller than 50ps is almost zero. The data gathered during the analysis is too voluminous to be presented here. The following paragraphs present smaller sets of data as an example.

Table I presents the SER (expressed in FIT) for an $AND2$, $NAND2$ and $XNOR2$ cells. As an example, it can be seen that for the $AND2$ cell, the soft-error rate of a SET with a Pulse-length of 50ps is equal to 51.1 FIT (Failure-In Time).

A comprehensive list of (PL, SER) for all the cells in the library (thirty-five cells) has been obtained and stored in a database. The next step is to build a logic gate-level model based on the SER calculations and pulse-lengths. Considering the fact that the key concept of simulation-based fault injection is to accelerate the occurrence rate of faults, then suppose that the initial number of fault injections in a design is β_{total}. The logic fault model for one cell can then be represented as Equation 1:

$$cell_{PL} = \frac{SER_{cell_{PL}} * \beta_{total}}{\sum_{i=50}^{175} SER_{cell_i}} \qquad \text{for all } cell \in lib. \quad (1)$$

where $cell_{PL}$ is the occurrence-number of pulses with a length of PL and $SER_{cell_{PL}}$ is the SER of a pulse with length of PL for a particular cell, labelled $cell$ (that can be directly extracted from the cell SER repository). So, $cell_{PL}$ values can be rapidly calculated for all the possible PL (six values), and for all the possible cells in the library (thirty-five cells) using any conventional mathematical tool.

As a result of Equation 1, the combination of $(Cell, PL, Cell_{PL})$ can be defined for all the possible cells and pulse-lengths. This information can be stored in a database for any desirable number of fault injections. Table II uses Equation 1 and the information of Table I to calculate the occurrence-number of each pulse-length for AND2, NAND2 and XNOR2 cells if the total number of fault injections (β_{total}) is 1000 (To simplify the calculations). This table shows that, e. g. for AND2 cell, 564 out of 1000 SET signals have the length of 50ps.

Starting from an initial number of fault injections, the logic simulator program picks up a cell in the synthesised gate-level design and based on the numbers calculated via Equation 1, those number of pulses with a specific pulse-length will be injected into that cell. Equation 2 shows a formula that takes into account the number of fault injections to calculate the occurrence-number of each pulse-length. The $end - point$ in Equation 2 is a point where increasing the number of fault injections will not be manageable any more to carry out experiments in a reasonable period of time.

TABLE II: Occurrence-rate of different PLs if number of fault injections is 1000

Cell	50ps	75ps	100ps	125ps	150ps	175ps
AND2	564	331	105	0	0	0
NAND2	639	316	43	1	0	0
XNOR2	429	371	187	12	1	0

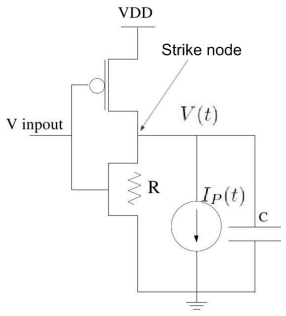

Fig. 1: SPICE representation of induced voltage in the strike node

$$cell_{PL} = \sum_{total=initial}^{end-point} \frac{SER_{cell_{PL}} * \beta_{total}}{\sum_{i=50}^{175} SER_{cell_i}} \text{ for all } cell \in lib. \quad (2)$$

IV. Analytical-based pulse-length determination technique

This section describes a logic pulse model in which its pulse-length has been determined based on the analytical model of a SET at the SPICE-level. The starting point is to define a SPICE representation of a SET. There are several models for such phenomenon in the literature. However the most commonly used representation is a transient pulse current ($I_P(t)$) inserted in the strike node, as shown in Figure 1. Part of this transient current comes from the node capacitance (c), the other part comes from V_{DD}. Several models have been used to formulate $I_P(t)$, however the most commonly accepted model is the double exponential pulse method [7], [8], given by Equation 3. This model has two timing parameters, γ_α and γ_β which are respectively the rising and falling time constants of the exponential equation.

$$I_P(t) = I_0 \left(e^{\frac{-t}{\gamma_\alpha}} - e^{\frac{-t}{\gamma_\beta}} \right) \quad (3)$$

where I_0 denotes the maximum charge collection current. The values of I_0, γ_α and γ_β are dependent on the used technological process and the particle of interest.

In the following paragraphs, an analytical calculation will be made to simplify the exponential behaviour of Equation 3 to determine the pulse-length of this equation based on the timing parameters of the node ($\gamma_\alpha, \gamma_\beta$). Parts of this calculation (Equations 4-6) come from reference [7].

From Figure 1, the following first-order differential equation can be derived:

$$C\frac{dV(t)}{dt} + \frac{V(t)}{R} = I_P(t) \tag{4}$$

where $V(t)$ is the voltage of the strike node and R is the equivalent resistance of the strike transistor. Solving this differential equation provides the voltage $V(t)$ at the strike node. This equation is given by:

$$V(t) = \frac{I_0\gamma_\alpha R}{\gamma_\alpha - RC}\left(e^{\frac{-t}{\gamma_\alpha}} - e^{\frac{-t}{RC}}\right) \tag{5}$$

We are interested to extract two parameters in Equation 5. First, t_{peak} at which the strike node voltage (or $V(t)$) reaches its peak value (V_{peak}), and the other parameter is $t_{de-active}$ at which the SET voltage will be de-activated due to conduction. It is defined that the threshold of de-activation is $V_{DD}/2$. Using mathematical optimization theory, the maximum value of t can be represented as:

$$t_{peak} = \frac{\ln\left(\frac{\gamma_\alpha}{RC}\right)\gamma_\alpha RC}{\gamma_\alpha - RC} \tag{6}$$

Solving Equation 5 for $V(t) = V_{DD}/2$ provides the value of $t_{de-active}$, as shown in Equation 7:

$$\begin{aligned} t_{de-active} = \frac{\ln\left(\frac{\gamma_\alpha}{RC}\right)\gamma_\alpha RC}{\gamma_\alpha - RC} - \\ RC\ln\left(\frac{\frac{V_{DD}}{2}}{V_{peak}}\right) - \gamma_\alpha \ln\left(\frac{\frac{V_{DD}}{2}}{V_{peak}}\right) \end{aligned} \tag{7}$$

Finally, as the pulse-length (PL) is the difference between $t_{de-active}$ and t_{peak}, one can derive:

$$PL = \beta.RC + \beta.\gamma_\alpha \tag{8}$$

where β is a constant number depends on the technology. Equation 8 shows the pulse-length as function of the strike-node parameters. It is important to mention that this model is only valid if the maximum amplitude of a SET is lower than the V_{peak}. In the following paragraphs, the asymptotic behaviour of Equation 8 will be explored to determine a simplified model for PL.

Suppose that RC is much greater than γ_α, then the pulse-length will be dominated by RC of the strike-transistor and the particle would not have sufficient energy to alter the output of the gate. This condition would be fulfilled if the strike-transistor changes its state (on to off or vice-versa) by an input stimulus. However, if γ_α is much greater than RC, then the pulse-length will be dominated by the energy of the particle of interest. In other words, as long as there is no activity on the strike node, i.e. no write activity, the pulse-length will be dominated by the particle, otherwise the pulse-length will be dominated by the normal transistor.

The described behaviour can be implemented in modern logic simulators via advanced programming languages. We have developed a VPI interface (Verilog Programming Interface), that imitates the described behaviour of Equation 8. So, if there is a new activity on the strike node (writing a new value), the impact of the particle would be neglected via conduction, otherwise the particle can change the state of that node. Fault injection experiments have been carried out using this model and its accuracy has been compared with the first fault model introduced in Section III, as well as other conventional pulse-lengths.

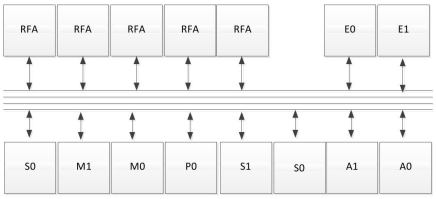

Fig. 2: The Xentium data-path architecture

V. The Xentium Processor

In order to assess the accuracy and applicability of the presented logic pulse representations, a high performance Very Large Instruction Word (VLIW) processor, the Xentium processor, has been selected. The Xentium processor which has been designed by Recore Systems [18] has been developed to be used in hostile environments, such as satellites. The detailed architecture of the data-path of the Xentium processor where fault injections have been carried out is shown in Figure 2. The data-path is designed based on a VLIW architecture that consists of ten execution units and five register files. Execution units M0 and M1 are multipliers, A0, A1, S0 and S1 are normal ALU, units C0 and P0 manage the program-counter while units E0 and E1 provide a fast memory access for the data-path. All these units are connected to each other via a high-speed bus. This paper uses a gate-level implementation of the Xentium processor, without any SEE precaution mechanism, to conduct fault injection experiments using different fault models.

VI. Logic gate-level fault experiments

To show the behaviour of different pulse-lengths in the rectangular SET fault model, four different sets of gate-level fault injections have been used. The first set of experiments uses the circuit-based extracted pulse-length represented by Equation 2. This pulse-length model takes into account the weighted contribution of each cell in the library as well as the system clock-period. So, it can be considered as the most accurate pulse-length model. The second set of experiments is based on the analytical pulse-length determination technique in Section IV, represented by Equation 8. This model takes into account the activities of each strike node to construct a pulse-length (which includes the system clock-period). In order to compare the efficiency of these two SET models, two other sets of fault injection experiments have been carried out using conventional pulse-length models. The third set of experiments uses the conventional determination technique of pulse-lengths for a SET where a constant pulse-length (100ps) is selected for all the pulses [10]. This model does take into account the characteristics of the technology, but not the contribution of each cell in the library. The fourth set of experiments uses a discrete logic SET pulse model that calculates the pulse-length based on an exponential distribution-function of the system clock-period [14]. This model takes into account the system clock-period, but not technology of implementation nor the contribution of each cell in the library.

A signal processing program, named Finite Impulse Response (FIR), has been used as a workload in all fault injection experiments. The Xentium processor receives all the required inputs (filter coefficients, input vector) from a text-file and produces the output vector in an output-text file. The required

TABLE III: Percentage of affected nets and the CPU-time for different number of fault injections

	number of fault injections (K)					
	0.5	1	2	4	8	16
% of affected nets	1	2	4	8	16	32
CPU-time (hours)	1.1	2.3	5	11	24	49

time to execute one run of the FIR program is about eight seconds (real time), so it is feasible to carry out a large number of experiments using this workload.
The types of failure on the processor can be classified as Silent-Data-Corruption (SDC) or Detected-Unrecoverable-Error (DUE) [19]. Since the Xentium processor does not have an error indicator signal one can not observe DUEs. As a result, the response of each injected fault has been classified in one of these categories:

1) $Silent-Data-Corruption, SDC$. This condition is met if the error propagates through the circuit without awareness of its occurrence by the system.
2) $Time-Out$ is the possibility that the processor stops its application without providing any meaningful data.
3) $Correct-behaviour$. the processor completes the application, even taking more computational time than the usual.

The results of the processor have been indicated in percent (%), e.g. if 10 out of 100 fault injections produce a Time-Out failure, the Time-Out failure sensitivity of the processor will be defined as 10%.
The fault injection has been carried out on all ten functional units of the Xentium processor (Figure 2). In each fault-injection experiment, one net within the functional units is automatically selected; Then the desired fault model is injected into the design and finally the results of the processor are compared with the correct values obtained from a Java-based fault-free simulator. The number of experiments has been increased from 500 (0.5K) to 16,000 (16K) experiments. It is worth mentioning that with 0.5K experiments, only 1% of the all nets in the Xentium data-path are affected. This rate is about 32% if the number of experiments reaches 16K. Table III shows the percentage of nets which are affected, as well as the required CPU-time (on a dual six-core Intel computer) if the number of fault injection grows from 0.5K to 16K.

VII. RESULTS

The behaviour of the Xentium processor for the rectangular SET model with four different pulse-length conditions has been depicted in the chart of Figure 3. The Y-axis in this chart shows the sensitivity of the processor for Silent-Data-Corruption (SDC), which means the percentage of injected faults that have been propagated into the main output. The X-axis is the number of fault injections which has been increased from 0.5K to 16K. The initial value of fault injections has been set to 0.5K and it has been increased until the required CPU-time to manage the experiments is still reasonable. The stop point for the number of injections was 16K because the required CPU-time to accomplish this number of fault injections already reached to more than forty-eight hours.

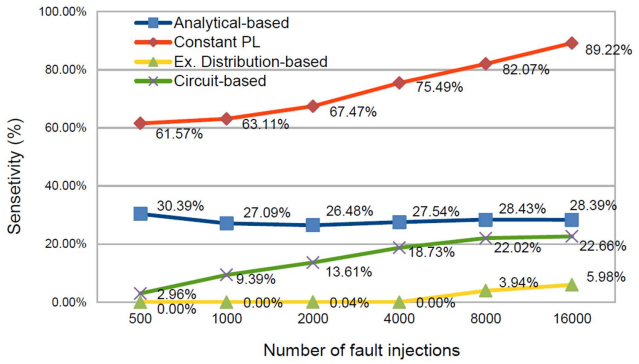

Fig. 3: Sensitivity of the Xentium processor for different fault models

The first observation from the chart of Figure 3 can be inferred about the general behaviour of the conventional fault models. It can be seen that the *constant pulse-length* model (constant PL) overestimates the contribution of SETs in the Xentium processor compare to the presented fault models; Because the contribution of different cells to produce a SET has not been considered in the *constant PL* model and as a consequence a very pessimistic model will be applied to the design. The *exponential distribution-based* model totally underestimates the contribution of SETs for a low volume of fault injections. So, a considerable amount of CPU time is still required to achieve a reasonable results while using this model.
One of the goal here is to find out how the convergence of the response of the processor is for each SET fault model. The sooner a fault model can reach a convergence-point, the better that fault model is equipped to explore a fast anticipation of the behaviour of the processor. To explore the convergence of each fault model, the theory of convergence of real numbers has been employed. By definition, the distance between two real numbers is the absolute value of their difference. For example, if a_i and a_j are two terms of a sequence $\{a_n\}$, the distance between a_i and a_j, denoted by $d(a_i, a_j)$ is defined as:

$$d(a_i, a_j) = |a_i - a_j| \tag{9}$$

One can say that $a_{convergence}$ is a *limit of sequence* $\{a_n\}$ if

$$\forall \epsilon > 0, \exists n_0 \in N : d(a_n, a_{convergence}) < \epsilon, \forall n > n_0 \tag{10}$$

if $a_{convergence}$ is a limit of sequence $\{a_n\}$, one could say that the sequence $\{a_n\}$ is a *convergent sequence* and it converges to $a_{convergence}$. In our experiments, $\{a_n\}$ is defined as the series of SDC sensitivities for each fault model, which can be extracted directly from the chart of Figure 3. As an example in the case of *circuit-based model* we have: $a_0 = 2.96\%$, $a_1 = 9.39\%$, $a_2 = 13.61\%$, $a_4 = 18.73\%$, $a_8 = 22.02\%$ and $a_{16} = 22.66\%$. Moreover, we assume that $a_{convergence}$ in each SET fault model is the SDC sensitivity of 16K number of fault injections for that specific SET fault model, or a_{16} (which also can be directly extracted from the chart of Figure 3). For example the $a_{convergence}$ of *circuit-based model* is assumed to be 22.66%. Our goal is to examine the possibility of finding a n_0 value, in Equation 10, while the ϵ value is relatively low. So as a result one can determine how soon fault injection results, $\{a_n\}$ converge to their final results, $a_{convergence}$ for

TABLE IV: Distance between the SDC sensitivity and the final SDC for each fault models

	number of fault injections (K)				
Fault model	0.5	1	2	4	8
Analytical-based	2	1.3	1.91	0.85	0.04
Constant PL	27.65	26.11	21.75	13.73	7.15
Ex. Distribution-based	5.98	5.98	5.94	5.98	2.04
Circuit-based	19.7	13.27	9.05	3.93	0.64

each specific SET fault model.

To achieve the mentioned goal, the distance d needs to be calculated for different numbers of fault injections in each fault model. A reasonably small d for the smallest number of fault injections (smallest n) is an indication to the speed of convergence of a sequence. Table IV shows the calculated d for each fault model in the case of a different number of fault injections. These numbers have been extracted directly by calculating the difference between the SDC sensitivity of a specific number of fault injections (the chart in the Figure 3) and the SDC sensitivity of 16K number of fault injections for the same fault model. for example, in the case of *circuit-based fault model*, a_1 is 9.39 while the $a_{convergence}$ is 22.66, which gives the $d = |a_1 - a_{convergence}| = 13.27$. All the numbers of Table IV have been calculated so.

Referring to Table IV, it can be seen that with assuming $\epsilon = 4$ in Equation 10, the *analytical-based* and *circuit-based* fault models show the best convergence if the number of experiments are equal and more than 4K (so the n and ϵ in the Equation 10 are equal to 4 and 4, respectively). This means that the results of the fault injections is stable for these two SET model if the number of fault injections is 4K or more. In the other words, the response of the Xentium processor will not be dependent on the number of fault injections for 4K number of fault injections and more. For the two conventional fault models, *constant PL* and *exponential distribution-based* models, the results are still deviating if the number of fault injections is even in the order of 8K, because the d is considerably greater than the other two fault models. This indicates that no interpretation can be made based on a small number of fault injections in these two SET models. One needs to increase the number of fault injections for these two conventional fault models (especially *constant PL model*) to reach the convergence.

Another interesting observation in Table IV can be seen by looking at the convergence of the *analytical-based* fault model for a very small number of fault injections (starting from 1K, $n = 1$). This means that one is able to have a very quick estimation of the processor sensitivity with regard to a SET even with a very low number of experiments (1K instead of 16K); Furthermore the required CPU-time to carry out 1K number of experiments is sixteen times smaller as compared to the required time to carry out all 16K fault injections.

VIII. CONCLUSIONS

Two approaches for the determination for pulse-length which can be used to build the rectangular SET logic model have been presented. The first method has been extracted from laser-based experiments along with a detailed transistor-level SET analysis tool. This model represents the most realistic model to anticipate a system response to SETs. The second determination technique that has been developed is based on the asymptotic behaviour of SETs in the SPICE model. We showed that these two models could provide a fast anticipation for the sensitivity of the system with regard to SET. These two fault models will contribute to solve the current challenge of developing/adopting EDA tools for fast and improved SER evaluation.

REFERENCES

[1] G. L. Ries, G. S. Choi, and R. K. Iyer, "Device-level transient fault modeling," *International Symposium on Fault-tolerant Computing*, pp. 86–94, 1994.

[2] A. Miele, C. Sandionigi, M. Ottavi, S. Pontarelli, A. Salsano, C. Metra, and et. al., "High-reliability fault tolerant digital systems in nanometric technologies: Characterization and design methodologies," *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, pp. 121–125, 2012.

[3] M. Nicolaidis, *Soft Errors in Modern Electronic Systems*. Cambridge: Springer, 2011.

[4] J. Arlat, Y. Crouzet, J. Karlsson, P. Folkesson, E. Fuchs, and G. H. Leber, "Comparison of physical and software implemented fault injection techniques," *IEEE Transactions on Computers*, vol. 25, no. 2, pp. 247–252, 2003.

[5] H. Ziade, R. Ayoubi, and R. Velazco, "A survey on fault injection techniques," *The International Arab Journal of Information Technology*, vol. 1, no. 2, pp. 171–185, 2004.

[6] L. Entrena, M. Garca-Valderas, R. Fernandez-Cardenal, A. Lindoso, M. P. Garca, and C. Lopez-Ongil, "Soft error sensitivity evaluation of microprocessors by multilevel emulation-based fault injection," *IEEE Transactions on Computers*, vol. 61, no. 3, pp. 313–323, 2012.

[7] G. Wirth, M. Vieira, and F. L. Kastensmidt, "Accurate and computer efficient modelling of single event transients in CMOS circuits," *IET Journal of Circuits Devices Systems*, vol. 1, no. 2, pp. 137–142, 2007.

[8] R. Naseer, Y. Boulghassoul, J. Draper, S. DasGupta, and A. Witulski, "Critical charge characterization for soft error rate modeling in 90nm SRAM," *IEEE Symp. on Circuits and Systems*, pp. 1897–1882, 2007.

[9] J. H. Barton, E. W. Czeck, Z. Z. Segall, and D. Siewiorek, "Fault injection experiments using FIAT," *IEEE Transactions on Computers*, vol. 39, no. 4, pp. 575–582, 1990.

[10] W. Sheng, L. Xiao, and Z. Mao, "Versatile and efficient techniques for speeding-up circuit level simulated fault-injection campaigns," *IEEE Pacific Rim International Symposium on Dependable Computing*, pp. 17–23, 2008.

[11] D. B. Limbrick and W. H. Robinson, "Characterizing single event transient pulse widths in an open-source cell library using SPICE," *IEEE Workshop on Silicon Errors in Logic-System*, 2012.

[12] G. Kanawati, N. Kanawati, and J. Abraham, "Ferrari: A flexible software-based fault and error injection system," *IEEE Transactions on Computers*, vol. 44, no. 2, pp. 248–260, 1995.

[13] H. Zarandi, S. G. Miremadi, and A. Ejlali, "Dependability analysis using a fault injection tool based on synthesizability of HDL models," *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 485–492, 2003.

[14] A. Rohani and H. G. Kerkhoff, "Rapid transient fault insertion in large digital systems,"*Journal of Microprocessors and microsystems*, vol. 37, no. 2, pp. 147–154, 2013.

[15] iRoC technologies 2012. [Online]. Available: http://www.iroctech.com/soft-error-tools/tfit-cell-level/

[16] Si2 2012. [Online]. Available: http://www.si2.org/openeda.si2.org/projects/nangatelib

[17] D. Alexandrescu, E. Costenaro, and M. Nicolaidis, "A practical approach to single event transients analysis for highly complex designs," *IEEE Symp. on Defect and Fault Tolerance in VLSI Systems*, pp. 155–163, 2011.

[18] Recore Systems 2011. [Online]. Available: http://www.recoresystems.com/technology/xentium-technology

[19] S. Mukherjee, *Architecture Design for Soft Errors*. ISBN: 978-0-12-369529-1: Elsevier Inc, 2008.